NuGet Resources Extractor - 1.0

Extension ID

com.castsoftware.dmtdotnetnugetresourcesextractor

What's new?

Please see **NuGet Resources Extractor - 1.0 - Release Notes**.

Description

The **NuGet Resources Extractor** provides the means to configure an automatic extraction of **NuGet package dependencies** from a NuGet repository for use during a .NET analysis. In other words, NuGet package based source code that resides in a simple local or nuget.org location. This extractor should be used when you want to extract packages (.nupkg) based source code that is stored in a NuGet repository. For example, when your .NET application contains **.csproj files** which have package references defined, you can use this extractor to extract those NuGet packages from the NuGet repository. Example package references in a .csproj are shown below:

```
<ItemGroup>
    <!-- ... -->
    <PackageReference Include="Contoso.Utility.UsefulStuff" Version="3.6.0" />
    <!-- ... -->
</ItemGroup>
```

packages.config

The content of the **packages.config** file (if this file is delivered and referenced via the **.csproj** file) is handled differently depending on the release of **AIP Core** you are using on your AIP Node to perform the analysis:

AIP Core 8.3.37	The Nuget Resources Extractor will identify the **packages.config** file and any packages defined in it will be identified and extracted. For example:
	`<?xml version="1.0" encoding="utf-8"?>`
	`<packages>`
	`  <package id="jQuery" version="3.1.1" targetFramework="net46" />`
	`  <package id="NLog" version="4.3.10" targetFramework="net46" />`
	`</packages>`

AIP Core 8.3.36	The NuGet Resources Extractor will identify the packages.config file but any packages defined in it will be ignored.

CAST AIP compatibility

AIP Console	1.18.x

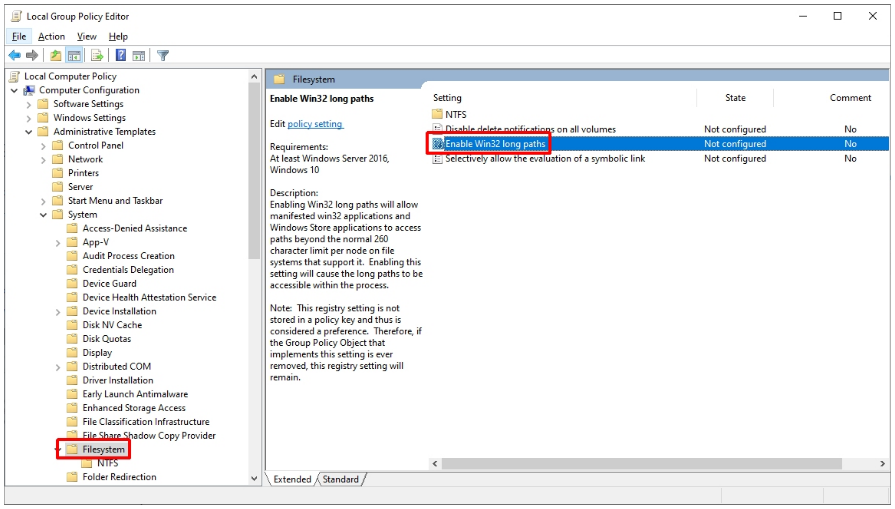
CAST AIP	8.3.27

Microsoft Windows Win32 long paths requirement

CAST highly recommends that **Win32 long paths** is enabled on the AIP Node on which the .NET analysis will be run: the extracted package dependencies are stored inside a folder called "**nugetPck**" which is located in the **Deployment** folder. Depending on where the Deployment folder is located, the path to the extracted package dependencies may exceed the default **260 character length**, which would cause the analyzer to fail to access the dependencies during an analysis. To enable long paths, use the **Local Group Policy Editor (gpedit)** on the AIP Node and drill down to:

Computer Configuration > Administrative Templates > System > Filesystem

Click to enlarge



Enable Win32 long paths

Edit policy setting

Requirements:
At least Windows Server 2016, Windows 10

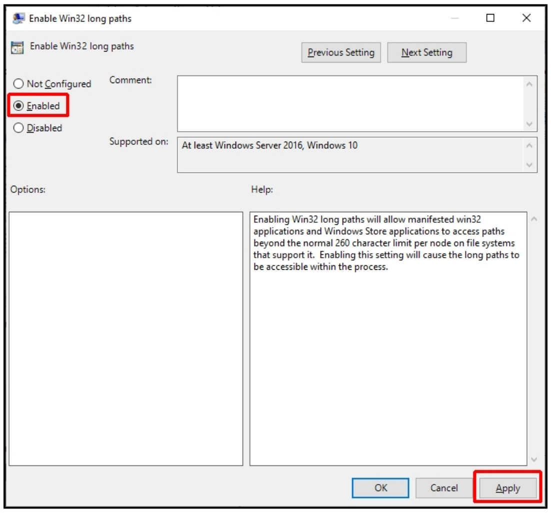
Description:
Enabling Win32 long paths will allow manifested win32 applications and Windows Store applications to access paths beyond the normal 260 character limit per node on file systems that support it. Enabling this setting will cause the long paths to be accessible within the process.

Note: This registry setting is not stored in a policy key and thus is considered a preference. Therefore, if the Group Policy Object that implements this setting is ever removed, this registry setting will remain.

Setting	State	Comment
NTFS		
Disable delete notifications on all volumes	Not configured	No
Enable Win32 long paths	Not configured	No
Selectively allow the evaluation of a symbolic link	Not configured	No

Then double click the **Enable Win32 long paths** option to enable the setting:

Click to enlarge



Enable Win32 long paths

○ Not Configured
● Enabled
○ Disabled

Supported on: At least Windows Server 2016, Windows 10

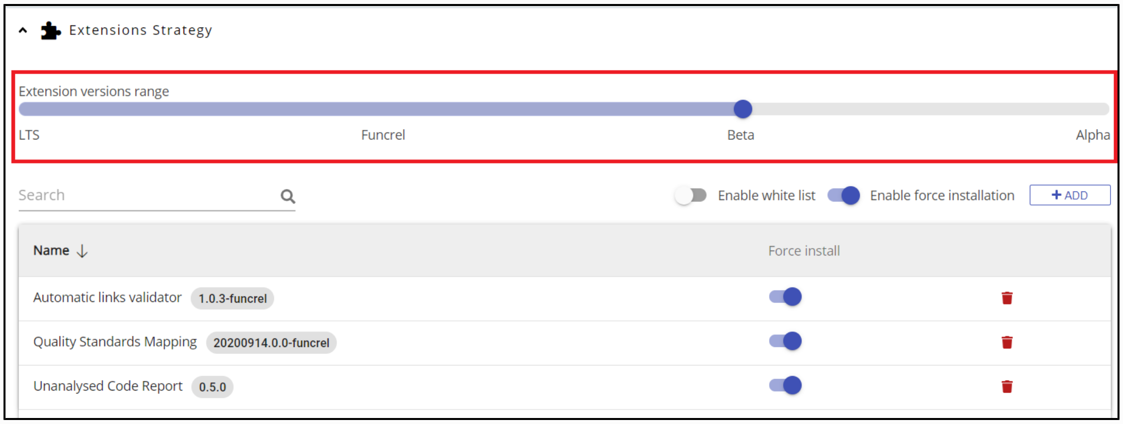
Help:
Enabling Win32 long paths will allow manifested win32 applications and Windows Store applications to access paths beyond the normal 260 character limit per node on file systems that support it. Enabling this setting will cause the long paths to be accessible within the process.

Using the extractor with AIP Console

Download and installation

The extractor will be automatically downloaded and installed in AIP Console when at least one **.csproj** file is delivered for analysis. You can also **manually install** the extension using the **Application - Extensions** interface.

The current extension is released in **beta** mode, therefore, to allow the extension to be automatically downloaded and installed when at least one **.csproj** file is delivered for analysis you will need to set the **extension strategy** version slider to include beta releases as shown below:



Name ↓	Force install
Automatic links validator 1.0.3-funcrel	●
Quality Standards Mapping 20200914.0.0-funcrel	●
Unanalysed Code Report 0.5.0	●

Usage

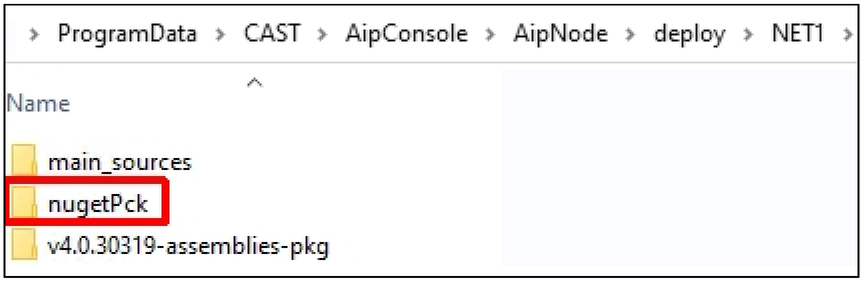
In AIP console this extractor is driven by the `%PROGRAMDATA%\CAST\AipConsole\AipNode\aip-node-app.properties` file attribute `scanner.nuget.repository`:

```
# HTTP V3 Nuget repository to download package dependencies https://api.nuget.org/v3/index.json or file system
like file://C:/Users/johndoe/.nuget/packages
scanner.nuget.repository=https://api.nuget.org/v3/index.json
```

By default this attribute is present and will accept two values:

- A **nuget.org** URL such as **https://api.nuget.org/v3/index.json** - this nuget.org url is provided in `aip-node-app.properties` by default.
- A **file system path** - the location input should start with protocol **file://** for example: `file://C:/Users/johndoe/.nuget/packages`

If a **.csproj** file is detected in the delivered source code, the extension will be downloaded and installed as part of the analysis process. If the **.csproj** file contains **package dependency references**, these references will automatically be accessed and extracted to a dedicated folder called **nugetPck** in the Application's **Deployment folder**:


› ProgramData › CAST › AipConsole › AipNode › deploy › NET1 ›

Name

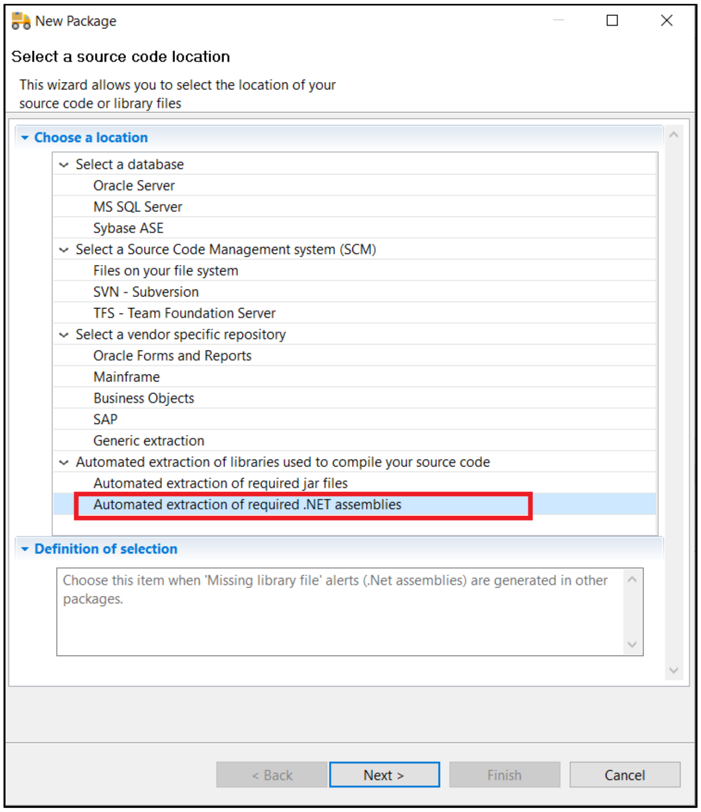
main_sources
nugetPck
v4.0.30319-assemblies-pkg

These references will then be used during the .NET analysis.

Using the extractor with the legacy CAST Delivery Manager Tool

The extractor can be accessed as follows:

Click to enlarge



New Package

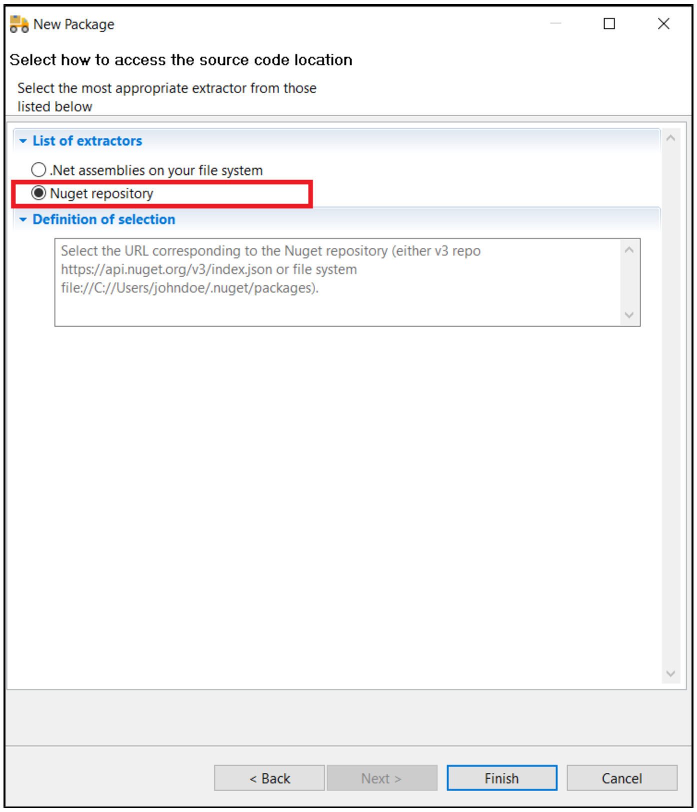
Select a source code location

This wizard allows you to select the location of your source code or library files

Definition of selection: Choose this item when 'Missing library file' alerts (.Net assemblies) are generated in other packages.

Choose a location
Select a database
Oracle Server
MS SQL Server
Sybase ASE
Select a Source Code Management system (SCM)
Files on your file system
SVN - Subversion
TFS - Team Foundation Server
Select a vendor specific repository
Oracle Forms and Reports
Mainframe
Business Objects
SAP
Generic extraction
Automated extraction of libraries used to compile your source code
Automated extraction of required jar files
Automated extraction of required .NET assemblies
Definition of selection
< Back
Next >
Finish
Cancel

Click to enlarge



New Package

Select how to access the source code location

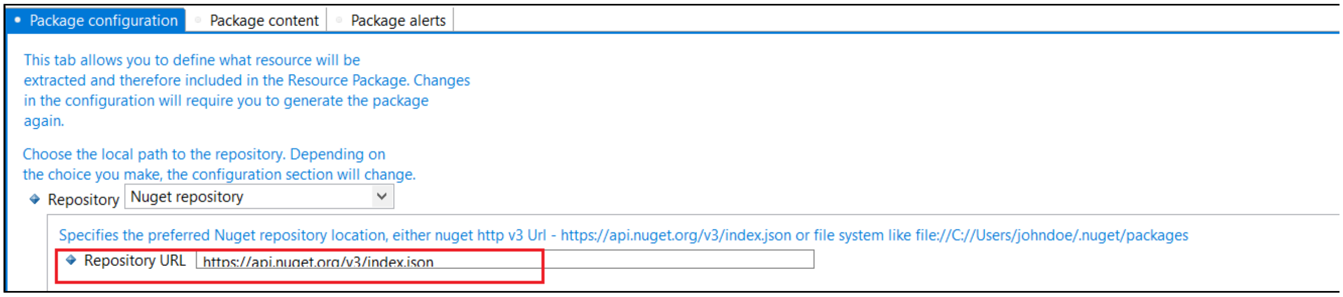
Select the most appropriate extractor from those listed below

List of extractors:

○ .Net assemblies on your file system
● Nuget repository

Definition of selection: Select the URL corresponding to the Nuget repository (either v3 repo https://api.nuget.org/v3/index.json or file system file://C://Users/johndoe/.nuget/packages).

Click to enlarge



This tab allows you to define what resource will be extracted and therefore included in the Resource Package. Changes in the configuration will require you to generate the package again.

Choose the local path to the repository. Depending on the choice you make, the configuration section will change.

Repository: Nuget repository

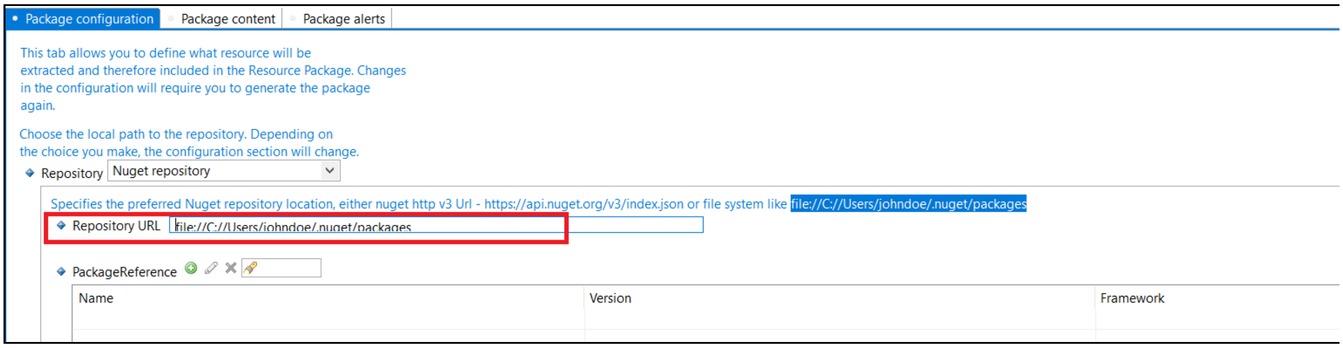
Specifies the preferred Nuget repository location, either nuget http v3 Url - https://api.nuget.org/v3/index.json or file system like file://C://Users/johndoe/.nuget/packages

Repository URL: https://api.nuget.org/v3/index.json

Enter the path to your NuGet repository, for example:

- A **nuget.org** URL such as **https://api.nuget.org/v3/index.json** - this nuget.org url is provided by default.
- A **file system path** - the location input should start with protocol **file://** for example: `file://C:/Users/johndoe/.nuget/packages`

Click to enlarge



This tab allows you to define what resource will be extracted and therefore included in the Resource Package. Changes in the configuration will require you to generate the package again.

Choose the local path to the repository. Depending on the choice you make, the configuration section will change.

Repository: Nuget repository

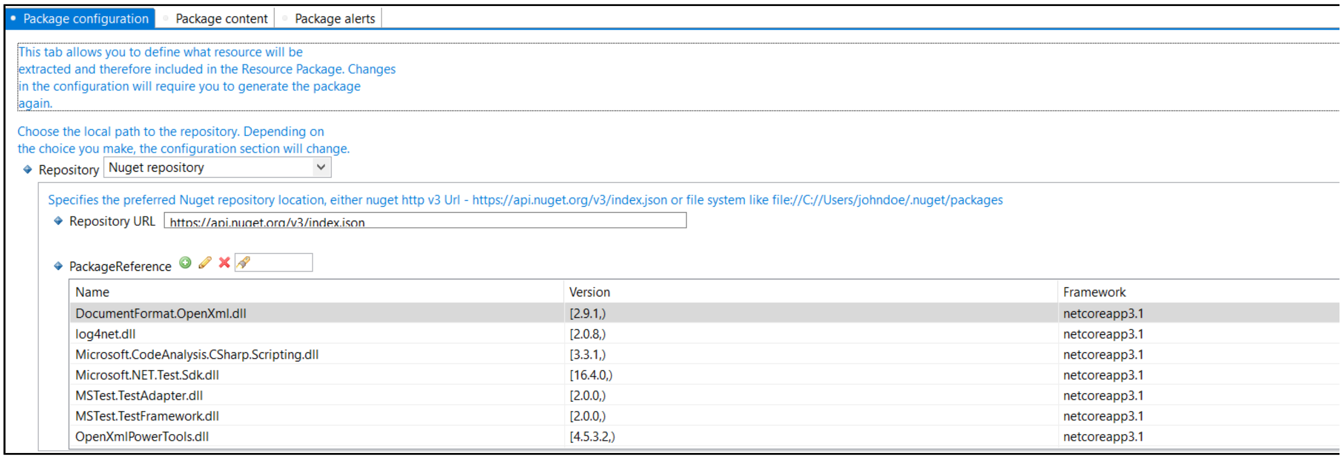
Specifies the preferred Nuget repository location, either nuget http v3 Url - https://api.nuget.org/v3/index.json or file system like file://C://Users/johndoe/.nuget/packages

Repository URL: file://C://Users/johndoe/.nuget/packages

PackageReference

Name	Version	Framework

When packaging is complete, the package dependencies identified and to be extracted will be displayed under **Package Reference** with name, version and framework:



This tab allows you to define what resource will be extracted and therefore included in the Resource Package. Changes in the configuration will require you to generate the package again.

Choose the local path to the repository. Depending on the choice you make, the configuration section will change.

Repository: Nuget repository

Specifies the preferred Nuget repository location, either nuget http v3 Url - https://api.nuget.org/v3/index.json or file system like file://C://Users/johndoe/.nuget/packages

Repository URL: https://api.nuget.org/v3/index.json

PackageReference

Name	Version	Framework
DocumentFormat.OpenXml.dll	[2.9.1,)	netcoreapp3.1
log4net.dll	[2.0.8,)	netcoreapp3.1
Microsoft.CodeAnalysis.CSharp.Scripting.dll	[3.3.1,)	netcoreapp3.1
Microsoft.NET.Test.Sdk.dll	[16.4.0,)	netcoreapp3.1
MSTest.TestAdapter.dll	[2.0.0,)	netcoreapp3.1
MSTest.TestFramework.dll	[2.0.0,)	netcoreapp3.1
OpenXmlPowerTools.dll	[4.5.3.2,)	netcoreapp3.1

Extraction messages

The following messages emitted by the extractor may appear during the source code extraction process

Format	Message Id	Message	Remediation
Error	com.castsoftware.dmt.extractor.dotnet.nuget.EmptyURL	Empty Url	Provide a valid repository URL.
Error	com.castsoftware.dmt.extractor.dotnet.nuget.UnexpectedRepository	Unexpected URL %URL%	Check the repository URL. It should be valid.
Error	com.castsoftware.dmt.extractor.dotnet.nuget.InstallationError	NuGet Not Installed false	Install the NuGet client tool. Follow https://docs.microsoft.com/en-us/nuget/install-nuget-client-tools.

Warning	com.castsoftware.dmt.extractor.dotnet.nuget.invalid URL	Invalid URL: %URL%	Provide a valid repository URL, either nuget.org (https://api.nuget.org/v3/index.json) or file system (file://C://Users/johndoe/.nuget/packages).
Warning	com.castsoftware.dmt.extractor.dotnet.nuget.extractPackages	No package artifact found!!!	Check the source discovery: packages should be identified as resource profile references in the format [packageName.dll][frameworkVersion] for each module.
Warning	com.castsoftware.dmt.extractor.dotnet.nuget.NugetCommandUtility	Output: %commandoutput% ERROR in setting source command: %Command%	Check the %commandoutput%.